

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-020308

(43)Date of publication of application : 21.01.2000

(51)Int.Cl. G06F 9/38
G06F 9/32

(21)Application number : 10-185206 (71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

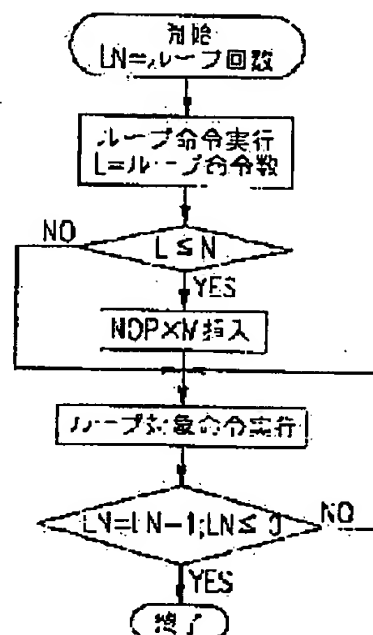
(22)Date of filing : 30.06.1998 (72)Inventor : YAMAZAKI MASAYUKI

(54) METHOD AND DEVICE FOR CONTROLLING PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a method and a device for controlling a program with which loop processing is performed at a high speed without limiting the minimum number of instructions desired to perform the loop processing even when the number of pipeline steps is increased by inserting the suitable number of no operation(NOP) instructions corresponding to the number of loop object instructions in the method and device for controlling the program with which the loop processing is performed for repeatedly executing plural instructions at an information processor for pipeline-processing the instructions.

SOLUTION: Concerning this program control method related to the loop processing for repeatedly executing plural instructions following a loop instruction at an information processor for pipeline-processing the instructions, this method is provided with a step for obtaining the number L of loop object instructions by decoding the said loop instruction, a step for executing the NOP instruction M times when the said number L of the loop object instructions is less than a prescribed number N, a step for executing the plural instructions and a step for discriminating whether the said step for executing the plural instructions is performed as many times as the prescribed number of times of the loop or not.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japanese Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-20308
(P2000-20308A)

(43) 公開日 平成12年1月21日 (2000.1.21)

(51) IntCl. ⁷	識別記号	F I	テマコード (参考)
G 0 6 F 9/38	3 3 0	G 0 6 F 9/38	3 3 0 E 5 B 0 1 3
9/32	3 3 0	9/32	3 3 0 B 5 B 0 3 3

審査請求 未請求 請求項の数 7 O L (全 13 頁)

(21) 出願番号 特願平10-185206
(22) 出願日 平成10年6月30日 (1998.6.30)

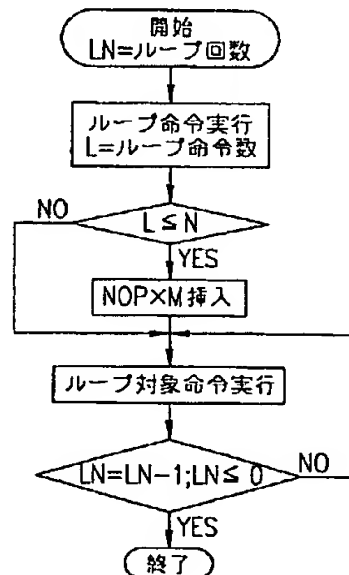
(71) 出願人 000005821
松下電器産業株式会社
大阪府門真市大字門真1006番地
(72) 発明者 山▲崎▼ 雅之
大阪府門真市大字門真1006番地 松下電器
産業株式会社内
(74) 代理人 100078282
弁理士 山本 秀策
Fターム (参考) 5B013 BB07
5B033 AA01 AA13 BE07 CA11

(54) 【発明の名称】 プログラム制御方法およびその装置

(57) 【要約】

【課題】 命令をパイプライン処理する情報処理装置において複数の命令を繰り返し実行させるループ処理のプログラム制御方法および装置に関して、ループ対象命令数に応じた適切な数のNOP命令を挿入することにより、パイプライン段数が増えてもループ処理したい命令の最小数を限定せずに、高速にループ処理するプログラム制御方法および装置を提供すること。

【解決手段】 本発明によるプログラム制御方法は、命令をパイプライン処理する情報処理装置において、ループ命令に続く複数の命令を繰り返し実行するループ処理に関するプログラム制御方法であって、前記ループ命令をデコードすることによりループ対象命令数Lを得るステップと、前記ループ対象命令数Lが所定の数N以下の場合にNOP命令をM回実行するステップと、前記複数の命令を実行するステップと、前記複数の命令を実行する前記ステップが所定のループ回数だけ行われたか否かを判定するステップと、を含む。



【特許請求の範囲】

【請求項 1】 命令をパイプライン処理する情報処理装置において、ループ命令に続く複数の命令を繰り返し実行するループ処理に関するプログラム制御方法であつて、

前記ループ命令をデコードすることによりループ対象命令数 L を得るステップと、

前記ループ対象命令数 L が所定の数 N 以下の場合に NOP 命令を M 回実行するステップと、

前記複数の命令を実行するステップと、

前記複数の命令を実行する前記ステップが所定のループ回数だけ行われたか否かを判定するステップと、

を含むプログラム制御方法。

【請求項 2】 前記判定するステップが、前記ループ命令を処理するステージのうち Q 段目のステージ以降において行われ、

$N = Q - 2$ であり、かつ、 $M = N$ である、

請求項 1 に記載のプログラム制御方法。

【請求項 3】 前記判定するステップが、前記ループ命令を処理するステージのうち Q 段目のステージ以降において行われ、

$N = Q - 2$ であり、かつ、 $M = N - L + 1$ である、

請求項 1 に記載のプログラム制御方法。

【請求項 4】 命令をパイプライン処理する情報処理装置において、ループ命令に続く複数の命令を繰り返し実行するループ処理の実行を制御する、プログラム制御装置であつて、

前記ループ命令をデコードすることによりループ対象命令数 L を出力し、前記ループ対象命令数 L が所定の数 N 以下の場合に NOP 挿入信号を M 回だけ出力するループ制御手段と、

前記ループ対象命令数 L と前記 NOP 挿入信号に従って、前記情報処理装置が前記ループ処理を実行するように実行アドレスを出力するループ実行手段と、

前記 NOP 挿入信号に従って、前記実行アドレスに対応する命令と NOP 命令のいずれかを出力する命令出力手段と、

を備えたプログラム制御装置。

【請求項 5】 前記ループ実行手段は、前記ループ命令を処理するステージのうち Q 段目のステージ以降において、前記複数の命令が所定のループ回数だけ行われたか否かを判定する判定手段を備え、

$N = Q - 2$ であり、かつ、 $M = N$ である、

請求項 4 に記載のプログラム制御装置。

【請求項 6】 前記ループ制御手段は前記ループ命令をデコードすることによりさらにループ回数を出力し、前記ループ実行手段は、前記ループ命令を処理するステージのうち Q 段目のステージ以降において、前記複数の命令が所定のループ回数だけ行われたか否かを判定する判定手段を備え、

$N = Q - 2$ であり、かつ、 $M = N - L + 1$ である、

請求項 4 に記載のプログラム制御装置。

【請求項 7】 前記ループ制御手段は前記ループ命令をデコードすることによりさらにループ信号を出力し、前記ループ実行手段は、

前記 NOP 挿入信号に従って、前記実行アドレスを生成する実行アドレス生成手段と、

前記ループ信号に従って、前記実行アドレスをスタートアドレスとして格納するスタートアドレス格納手段と、

10 前記ループ信号に従って、前記実行アドレスに前記ループ対象命令数を加算したエンドアドレスを格納するエンドアドレス格納手段と、

前記実行アドレスと前記エンドアドレスが一致した場合に、前記複数の命令が所定のループ回数だけ行われたか否かを判定し、前記判定に従って分岐信号を出力するループ分岐判定手段と、

前記分岐信号に従って、前記実行アドレスと前記スタートアドレスのいずれかを出力するアドレス出力手段とを備えている、請求項 4 に記載のプログラム制御装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、命令をパイプライン処理する情報処理装置におけるループ処理のプログラム制御方法およびその装置に関する。

【0002】

【従来の技術】 命令をパイプライン処理する情報処理装置において、従来から複数の命令を繰り返し実行させるループ命令がある。このループ命令は、ループ対象となる命令の数（以下ループ対象命令数と表記）を指定することにより、ループ対象の命令を指定した回数繰り返し実行するループ処理を行うことを指示する。

【0003】 図 10 は、3 段パイプラインの情報処理装置におけるループ処理の動作を示す。いま、ループ対象命令が格納されている先頭の番地に分岐するか否かを判定することができるのは、現在実行中のループ処理を指示したループ命令を処理するステージの EX ステージ以降であると仮定する。この仮定の下では、ループ対象命令が格納されている先頭の番地に分岐するか否かを判定するまでに、ループ命令に続くいくつかの命令が実行されてしまうことになる。図 10 に示されるような、IF ステージ、DE ステージ、EX ステージから構成される 3 段パイプラインの場合には、ループ命令の EX ステージはループ命令の次の命令 1 の DE ステージ、命令 1 の次の命令 2 の IF ステージに相当する。よって、命令 1 と命令 2 の 2 命令の処理が開始される前には、ループ命令による分岐判定をすることはできない。その結果、ループ命令が実行される場合には、ループ命令の直後の命令 1 と命令 2 は常に実行されることとなる。そのため、ループ処理の対象にできる命令の最小数（以下「最小ループ対象命令数」と表記する。）は 2 である。もし、ル

ープ処理の対象としたい命令が 1 命令である場合には、ループ命令の次に無処理命令（以下「NOP 命令」と表記する。）をコンパイラ等が挿入しておく必要がある。しかし、ループ命令の次に NOP 命令を挿入した場合、NOP 命令がループ対象命令に含まれるため、ループ処理を行う回数だけ無駄な NOP 命令が実行されることになる。

【0004】図 11 は、4 段パイプラインの情報処理装置におけるループ処理の動作を示す。上述した 3 段パイプラインの情報処理装置と同じ仮定の下では、命令 1 と命令 2 と命令 3 の 3 命令の処理が開始される前には、ループ命令による分岐判定をすることはできない。その結果、ループ命令が実行される場合には、ループ命令の直後の命令 1 と命令 2 と命令 3 の 3 命令は常に実行されることとなる。そのため、最小ループ対象命令数は 3 である。もし、ループ処理の対象としたい命令が 1 命令である場合には、ループ命令の次に NOP 命令を 2 回挿入しておく必要がある。ループ命令の次に NOP 命令を 2 回挿入した場合、ループ処理を行う回数の 2 倍の回数だけ無駄な NOP 命令が実行されることになる。

【0005】このような無駄な NOP 命令の実行を回避するために、ループ命令を複数サイクル命令にする方法がある。すなわち、ループ命令がデコードされると、ループ命令の直後にハードウェアによって NOP 命令を挿入することにより、ループ命令を処理に複数サイクルを要する命令として取り扱う方法である。NOP 命令を挿入する回数は、最小ループ対象命令数と、ループ命令による分岐判定を行うことが可能になるステージが当該ループ命令を処理するパイプラインステージのうちの何番目のステージであるかに依存する。例えば、最小ループ対象命令数を 1 とすると、上述した 3 段パイプラインの例では 1 回であり、4 段パイプラインの例では 2 回となる。上記の方法により、ループ命令の直後に NOP 命令をコンパイラ等が挿入しておく必要がなくなり、ループ処理を行う度に無駄な NOP 命令が実行されることがなくなる。ループ命令を複数サイクル命令にする方法は、例えば、日本テキサス・インスツルメンツ株式会社の TMS320C30 ユーザーズ・マニュアルの 7-2 頁、7-3 頁、11-153 頁、11-154 頁に記載されている。しかし、この方法を用いた場合には、ループ対象命令数に関係なく NOP 命令が挿入されるため、ループ対象命令数が最小ループ対象命令数より多い場合にも無駄な NOP 命令が挿入されるという問題があった。

【0006】

【発明が解決しようとする課題】 上述したように、従来の情報処理装置におけるループ処理では、ループ命令による分岐判定を行うことが可能になるステージが当該ループ命令を処理するパイプラインステージのうちの何番目のステージであるかに依存して、最小ループ対象命令数が限定される。その結果、ループ処理の対象としたい

命令の数が最小ループ対象命令数より少ない場合には、無駄な NOP 命令を挿入しておく必要があった。

【0007】また、ループ命令を複数サイクル命令とした場合、コンパイラ等が無駄な NOP 命令を挿入しておく必要はなくなるが、ループ命令を実行する毎に常に NOP 命令がハードウェアによって挿入されるため、ループ処理の対象としたい命令の数が最小ループ対象命令数より多い場合には無駄な NOP 命令が挿入されることになっていた。

10 【0008】本発明は以上の点に鑑み、命令をパイプライン処理する情報処理装置において複数の命令を繰り返し実行させるループ処理のプログラム制御方法および装置に関して、ループ対象命令数に応じた適切な数の NOP 命令を挿入することにより、パイプライン段数が増えてもループ処理したい命令の最小数を限定せずに、高速にループ処理するプログラム制御方法およびその装置を提供することを目的とする。

【0009】

20 【課題を解決するための手段】 本発明によるプログラム制御方法は、命令をパイプライン処理する情報処理装置において、ループ命令に続く複数の命令を繰り返し実行するループ処理に関するプログラム制御方法であって、前記ループ命令をデコードすることによりループ対象命令数 L を得るステップと、前記ループ対象命令数 L が所定の数 N 以下の場合に NOP 命令を M 回実行するステップと、前記複数の命令を実行するステップと、前記複数の命令を実行する前記ステップが所定のループ回数だけ行われたか否かを判定するステップと、を含む。

30 【0010】前記判定するステップは、前記ループ命令を処理するステージのうち Q 段目のステージ以降において行われ、 $N=Q-2$ であり、かつ、 $M=N$ であってもよい。

【0011】前記判定するステップは、前記ループ命令を処理するステージのうち Q 段目のステージ以降において行われ、 $N=Q-2$ であり、かつ、 $M=N-L+1$ であってもよい。

40 【0012】本発明によるプログラム制御装置は、命令をパイプライン処理する情報処理装置において、ループ命令に続く複数の命令を繰り返し実行するループ処理の実行を制御する、プログラム制御装置であって、前記ループ命令をデコードすることによりループ対象命令数 L を出力し、前記ループ対象命令数 L が所定の数 N 以下の場合に NOP 挿入信号を M 回だけ出力するループ制御手段と、前記ループ対象命令数 L と前記 NOP 挿入信号に従って、前記情報処理装置が前記ループ処理を実行するように実行アドレスを出力するループ実行手段と、前記 NOP 挿入信号に従って、前記実行アドレスに対応する命令と NOP 命令のいずれかを出力する命令出力手段と、を備えている。

50 【0013】前記ループ実行手段は、前記ループ命令を

処理するステージのうちQ段目のステージ以降において、前記複数の命令が所定のループ回数だけ行われたか否かを判定する判定手段を備え、 $N=Q-2$ であり、かつ、 $M=N$ であってもよい。

【0014】前記ループ実行手段は、前記ループ命令を処理するステージのうちQ段目のステージ以降において、前記複数の命令が所定のループ回数だけ行われたか否かを判定する判定手段を備え、 $N=Q-2$ であり、かつ、 $M=N-1+1$ であってもよい。

【0015】前記ループ制御手段は前記ループ命令をデコードすることによりさらにループ信号を出力し、前記ループ実行手段は、前記NOP挿入信号に従って、前記実行アドレスを生成する実行アドレス生成手段と、所定のループ信号に従って、前記実行アドレスをスタートアドレスとして格納するスタートアドレス格納手段と、前記ループ信号に従って、前記実行アドレスに前記ループ対象命令数を加算したエンドアドレスを格納するエンドアドレス格納手段と、前記実行アドレスと前記エンドアドレスが一致した場合に、前記複数の命令が前記ループ回数だけ行われたか否かを判定し、前記判定に従って分岐信号を出力するループ分岐判定手段と、前記分岐信号に従って、前記実行アドレスと前記スタートアドレスのいずれかを出力するアドレス出力手段とを備えていてもよい。

【0016】

【発明の実施の形態】以下、図面を参照して本発明の実施の形態を説明する。

【0017】（実施形態1）図1は、本発明によるプログラム制御装置7の構成を示す。プログラム制御装置7を含む情報処理装置は、パイプライン動作により、メモリ2に格納されたプログラムを実行する。プログラム制御装置7は、ループ実行部1と、NOP生成部3と、ループ制御部4とを含む。プログラム制御装置7を含む情報処理装置はさらに、デコーダ5と、演算部6とを含む。プログラム制御装置7から出力された命令は、デコーダ5によってデコードされることにより、演算部6において実行される。

【0018】ループ実行部1は、ループ処理の実行に必要なメモリ2上のアドレスを出力する。すなわち、ループ実行部1は、入力されたNOP挿入信号13とループ命令信号14とループ対象命令数15に応じて、第1のアドレス10を出力する。メモリ2は、プログラム制御装置7を含む情報処理装置が実行するプログラムを格納する。メモリ2は、入力された第1のアドレス10上にある命令を、第1の命令コード11として出力する。NOP生成部3は、メモリ2に格納されたプログラム中の複数の命令の間に必要に応じてNOP命令を挿入し、プログラム制御装置7を含む情報処理装置が次に実行すべき命令を出力する。NOP生成部3は、NOP挿入信号13に従って、入力された第1の命令コード11とNO

P生成部3が生成するNOP命令のいずれかを第2の命令コード12として出力する。ループ制御部4はループ処理を実行する場合に、プログラム制御装置7を制御する。ループ制御部4は、入力された第2の命令コード12をデコードすることにより、NOP挿入信号13とループ命令信号14とループ対象命令数15とを出力する。デコーダ5は入力された第2の命令コード12をデコードすることにより、演算部6を制御する制御信号を出力する。演算部6は、デコーダ5から出力された制御信号に従って、加算、減算等の各種の演算を実行する。

【0019】NOP挿入信号13の値は、第2の命令コード12としてNOP命令を出力しない場合には「0」であり、第2の命令コード12としてNOP命令を出力する場合には「1」である。ループ命令信号14の値は、ループ制御部4に入力された第2の命令コード12がループ命令でない場合には「0」であり、ループ制御部4に入力された第2の命令コード12がループ命令である場合には「1」である。

【0020】図2は本発明によるプログラム制御装置7に含まれるNOP生成部3の構成を示す。NOP生成部3は、命令レジスタ101と、第1のマルチプレクサ102とを含む。命令レジスタ101は、NOP挿入信号13の値が「0」のとき、入力された第1の命令コード11を格納する（ラッチする）。命令レジスタ101は、NOP挿入信号13の値が「1」のとき、その格納している命令を保持する。また、命令レジスタ101は、現在格納している命令を命令レジスタ出力112として出力する。第1のマルチプレクサ102は、入力された命令レジスタ出力112とNOPコード111のいずれかを選択し第2の命令コード12として出力する。第1のマルチプレクサ102は、NOP挿入信号13の値が「0」のとき、命令レジスタ出力112の内容を第2の命令コード12として出力する。第1のマルチプレクサ102は、NOP挿入信号13の値が「1」のとき、NOPコード111を第2の命令コード12として出力する。

【0021】図3は本発明によるプログラム制御装置7に含まれるループ実行部1の構成を示す。ループ実行部1は、アドレスレジスタ201と、第2のマルチプレクサ202と、スタートアドレスレジスタ203と、第1の加算器204と、第2の加算器205と、エンドアドレスレジスタ206と、第1の比較器207と、ループカウンタ208と、論理積ゲート209とを含む。

【0022】アドレスレジスタ201と、第2のマルチプレクサ202と、スタートアドレスレジスタ203と、第1の加算器204は、第1のアドレス10としてメモリ2に出力する実行アドレスを生成する。アドレスレジスタ201は、NOP挿入信号13の値が「0」のとき、入力された第2のマルチプレクサ出力213をラッチする。アドレスレジスタ201は、NOP挿入信号

7

13の値が「1」のとき、その格納しているアドレスを保持する。アドレスレジスタ201は、現在格納しているアドレスを第1のアドレス10として出力する。第2のマルチプレクサ202は、入力された第1の加算器出力212とスタートアドレス214のいずれかを選択し第2のマルチプレクサ出力213として出力する。第2のマルチプレクサ202は、分岐信号219の値が

「0」のとき、第1の加算器出力212の内容を第2のマルチプレクサ出力213として出力する。第2のマルチプレクサ202は、分岐信号219の値が「1」のとき、スタートアドレス214の内容を第2のマルチプレクサ出力213として出力する。スタートアドレスレジスタ203は、ループ処理の対象となる1つ以上の命令の最初アドレスであるスタートアドレスを格納する。スタートアドレスレジスタ203は、ループ命令信号14の値が「0」のとき、その格納しているアドレスを保持する。スタートアドレスレジスタ203は、ループ命令信号14の値が「1」のとき、入力された第1のアドレス10をラッチする。また、スタートアドレスレジスタ203は、現在格納しているアドレスをスタートアドレス214として出力する。第1の加算器204は、入力された第1のアドレス10に一定の値を加算して（インクリメントして）得られたアドレスを、第1の加算器出力212として出力する。

【0023】第2の加算器205と、エンドアドレスレジスタ206は、ループ処理の対象となる1つ以上の命令の最後のアドレスであるエンドアドレスを生成し保持する。第2の加算器205は、入力された第1のアドレス10にループ対象命令数15を加算して得られたアドレスを、第2加算器出力215として出力する。エンドアドレスレジスタ206は、エンドアドレスを格納する。エンドアドレスレジスタ206は、ループ命令信号14の値が「0」のとき、その格納しているアドレスを保持する。エンドアドレスレジスタ206は、ループ命令信号14の値が「1」のとき、入力された第2の加算器出力215をラッチする。また、エンドアドレスレジスタ206は、現在格納しているアドレスをエンドアドレス216として出力する。

【0024】第1の比較器207と、ループカウンタ208と、論理積ゲート209は、ループ命令による分岐判定を行う。第1の比較器207は、入力されたエンドアドレス216と第1のアドレス10を比較し、比較の結果に従ってエンド信号217を出力する。第1の比較器207は、エンドアドレス216と第1のアドレス10が一致しないとき、エンド信号217の値を「0」とする。第1の比較器207は、エンドアドレス216と第1のアドレス10が一致するとき、エンド信号217の値を「1」とする。ループカウンタ208は、ループ処理の対象となる1つ以上の命令が実行された回数である実行済ループ回数をカウントし、カウントの結果に応

8

じてゼロ信号218を出力する。ループカウンタ208は、エンド信号217が入力されるとその保持する残りループ回数から1を減算する。残りループ回数が0でないときループカウンタ208は、ゼロ信号218の値を「0」とする。残りループ回数が0になる、すなわち実行済ループ回数が指定されたループ回数に達すると、ループカウンタ208は、ゼロ信号218の値を「1」とする。論理積ゲート209は、入力されたエンド信号217とゼロ信号218との論理積（AND）を取り、分岐信号219として出力する。すなわち、論理積ゲート209は、エンド信号217とゼロ信号218のいずれかの値が「0」のときは、分岐信号219の値を「0」とする。論理積ゲート209は、エンド信号217とゼロ信号218の値がともに「1」のときは、分岐信号219の値を「1」とする。

【0025】なお、ループ回数をループカウンタ208に設定する方法には以下の3つの方法がある。（1）ループ命令を実行する前に別の命令によって設定する。

（2）ループ命令により、ループ命令の命令コード中に含まれているループ回数を設定する。（3）ループ命令により、演算器にある汎用レジスタ等のレジスタからループ回数を転送して設定する。

【0026】図4は本発明の実施形態1のプログラム制御装置7に含まれるループ制御部4の構成を示す。ループ制御部4は、デコーダ301と、第2の比較器302と、論理積ゲート303と、レジスタ304と、論理和ゲート305を含む。デコーダ301は、入力された第2の命令コード12をデコードし、デコードの結果に応じてループ命令信号14とループ対象命令数15を出力する。デコーダ301は、デコードした命令がループ命令でないときは、ループ命令信号14の値を「0」とする。デコーダ301は、デコードした命令がループ命令であるときは、ループ命令信号14の値を「1」とし、ループ命令のオペランドであるループ対象命令数をループ対象命令数15に出力する。第2の比較器302は、入力されたループ対象命令数15と数値「2」を比較し、比較の結果を第2の比較器出力313として出力する。第2の比較器302は、ループ対象命令数15の値が「2」より大きいときは、第2の比較器出力313の値を「0」とする。第2の比較器302は、ループ対象命令数15の値が「2」以下であるときは、第2の比較器出力313の値を「1」とする。論理積ゲート303は、入力された第2の比較器出力313とループ命令信号14との論理積である、論理積出力314を出力する。すなわち、論理積ゲート303は、第2の比較器出力313とループ命令信号14のいずれかの値が「0」のときは、論理積出力314の値を「0」とする。論理積ゲート303は、第2の比較器出力313とループ命令信号14の値がともに「1」のときは、論理積出力314の値を「1」とする。

【0027】レジスタ304は、入力された論理積出力314をラッチする。また、レジスタ304は、現在格納している値をレジスタ出力315として出力する。論理和ゲート305は、論理積出力314とレジスタ出力315との論理和（OR）をNOP挿入信号13として出力する。すなわち、論理和ゲート305は、論理積出力314とレジスタ出力315の値がともに「0」のときは、NOP挿入信号13の値を「0」とする。論理和ゲート305は、論理積出力314とレジスタ出力315のいずれかの値が「1」のときは、NOP挿入信号13の値を「1」とする。

【0028】以下、図1～図4および図6を参照して、本発明によるプログラム制御装置7の動作を説明する。

【0029】（命令のシーケンシャル実行）まず、命令をシーケンシャルに実行する場合のプログラム制御装置7の動作を説明する。命令をシーケンシャルに実行する場合には、NOP挿入信号13は「0」である。その結果、ループ実行部1は、順次インクリメントされた第1のアドレス10を出力する。メモリ2は、順次インクリメントされた第1のアドレス10が入力されることにより、メモリ2に格納された命令を順次第1の命令コード11として出力する。NOP生成部3は、NOP挿入信号13が「0」であるため、NOP命令を挿入せずに、入力された第1の命令11をそのまま第2の命令コード12として出力する。NOP生成部3から出力された第2の命令コード12は、デコーダ5によりデコードされ、演算部6において実行される。このように、メモリ2に格納された命令がプログラム制御装置7を含む情報処理装置においてシーケンシャルに実行される。

【0030】命令をシーケンシャルに実行する場合のループ実行部1の動作を以下に説明する。上述のように、命令をシーケンシャルに実行する場合にはNOP挿入信号13は「0」である。NOP挿入信号13が「0」であるとき、アドレスレジスタ201は第2のマルチプレクサ出力213をラッチし、第1のアドレス10として出力する。第1の加算器204は、入力された第1のアドレス10をインクリメントし、第1の加算器出力212として出力する。第2のマルチプレクサ202は、第1の加算器出力212を選択し、第2のマルチプレクサ出力213として出力する。以上の動作を繰り返す事により第1のアドレス10は順次インクリメントされていく。

【0031】また、NOP生成部3においては、NOP挿入信号13が「0」であるとき、命令レジスタ101は第1の命令コード11をラッチし、命令レジスタ出力112として出力する。NOP挿入信号13が「0」であるとき、第1のマルチプレクサ102は、命令レジスタ出力112を選択し、第2の命令コード12として出力する。

【0032】ループ制御部4に含まれるデコーダ301

は第2の命令コード12をデコードする。第2の命令コード12がループ命令でなければ、ループ信号14の値を「0」とし、ループ対象命令数15は出力しない。ループ信号14の値が「0」である限り、論理積ゲート303の出力である論理積出力314の値は「0」であり続ける。そのため、レジスタ304の値は「0」となり、レジスタ出力315の値も「0」となる。論理積出力314とレジスタ出力315の値がともに「0」であるため、論理和ゲート305の出力であるNOP挿入信号13の値は「0」となる。値「0」を有するNOP挿入信号13がループ実行部1に入力されることにより、上述の動作が繰り返される。

【0033】以上の動作を繰り返す事で、メモリ2に格納された命令がプログラム制御装置7を含む情報処理装置によってシーケンシャルに実行されることとなる。

【0034】（ループ命令の実行）次に、ループ命令を実行する場合のプログラム制御装置7の動作を説明する。プログラム制御装置7を含む情報処理装置が命令をシーケンシャルに実行している状態で、NOP生成部3によって第2の命令コード12としてループ命令が出力されたと仮定する。ループ制御部4に含まれるデコーダ301は、入力された第2の命令コード12がループ命令であると、ループ命令信号14を「1」にし、ループ対象命令数15を出力する。

【0035】ループ命令信号14が「1」になると、ループ実行部1においてスタートアドレスレジスタ203は、第1のアドレス10をラッチする。また、第2の加算器205が第1のアドレス10とループ対象命令数15を加算し、第2の加算器出力215として出力する。ループ命令信号14は「1」であるから、エンドアドレスレジスタ206は第2の加算器出力215をラッチする。このように、ループ実行部1において、スタートアドレスとエンドアドレスがラッチされる。

【0036】ループ制御部4の動作は、ループ対象命令数15の値が「2」より大きいのか、「2」以下であるかによって異なる。ループ制御部4において、第2の比較器302は、入力されたループ対象命令数15の値が「2」より大きい場合には第2の比較器出力313の値を「0」とする。その結果、論理積出力314の値は「0」のまま変化せず、NOP挿入信号13の値も「0」のまま変化しない。従って、ループ対象命令数15の値が「2」より大きい場合には、NOP生成部3においてNOP命令の挿入は行われない。

【0037】入力されたループ対象命令数15の値が「2」以下の場合は、第2の比較器302は、第2の比較器出力313の値を「1」にする。デコーダ301がループ命令をデコードした時点におけるループ命令信号14の値は「1」であるから、論理積ゲート303は、第2の比較器出力313の値が「1」になると論理積出力314の値を「1」とする。論理積出力314の値が

11

「1」となる結果、レジスタ304は値「1」をラッチする。また、論理積出力314の値が「1」となる結果、論理和ゲート305は出力するNOP挿入信号13の値を「1」にする。

【0038】NOP挿入信号13が「1」になると、NOP生成部3において、命令レジスタ101は格納している命令をそのまま保持し、第1のマルチプレクサ102はNOPコード111を選択する。その結果、第1のマルチプレクサ102が出力する第2の命令コード12はNOP命令となる。また、ループ実行部1に含まれるアドレスレジスタ201は、NOP挿入信号13が

「1」になると、格納しているアドレスをそのまま保持する。このように、NOP挿入信号13が「1」になると、命令レジスタ101およびアドレスレジスタ201は更新を停止し、NOP生成部3から第2の命令コード12としてNOP命令が出力される。

【0039】デコーダ301に第2の命令コード12としてNOP命令が入力されると、NOP命令はループ命令ではないため、ループ命令信号14の値は「0」となる。このデコーダ301に入力されたNOP命令は、プログラム制御装置7を含む情報処理装置においてループ命令の次に実行される命令である。ループ命令信号14の値が「0」になると、論理積出力314の値は「0」となる。しかし、レジスタ304がその保持する値

「1」をレジスタ出力315として出力するため、論理和ゲート305の出力するNOP挿入信号13の値は再度「1」となる。なお、レジスタ304は、値「1」をレジスタ出力315として出力した後、論理積出力314の値「0」をラッチする。

【0040】NOP挿入信号13の値が再び「1」であるため、命令レジスタ101およびアドレスレジスタ201は更新を停止したままとなり、NOP生成部3から第2の命令コード12としてNOP命令が再度出力される。デコーダ301に第2の命令コード12としてNOP命令が再度入力されると、NOP命令はループ命令ではないため、ループ命令信号14の値は「0」となる。このデコーダ301に再度入力されたNOP命令は、プログラム制御装置7を含む情報処理装置においてループ命令の2つ後に実行される命令である。ループ命令信号14の値が「0」になると、論理積出力314の値は「0」となる。レジスタ304はその保持する値「0」をレジスタ出力315として出力するため、論理和ゲート305の出力するNOP挿入信号13の値は今度は「0」となる。

【0041】NOP挿入信号13の値が「0」になると、NOP生成部3に含まれる第1のマルチプレクサ102は命令レジスタ出力112を選択する。すなわち、NOP生成部3は、NOP命令の挿入を行わない。

【0042】以上のように、ループ対象命令数15の値が「2」以下である場合には、NOP挿入信号13の値

12

が2回「1」となる。その結果、NOP生成部3においてNOP命令が2回挿入される。

【0043】(ループ処理中の実行) 次に、ループ処理中のプログラム制御装置7の動作を説明する。上述したように、ループ対象命令数が2より大きいければNOP挿入信号13の値が「0」のまま変化せず、ループ対象命令数が2以下であればNOP命令挿入後にNOP挿入信号13の値は「0」となる。

【0044】ループ対象命令数は1以上であるため、ループ命令の実行直後には、ループ実行部1において、第1のアドレス10の値とエンドアドレス216の値は一致しない。ループ対象命令数が2以下である場合の、挿入されたNOP命令の実行直後も同様である。従って、ループ命令の実行直後には、第1の比較器207は出力するエンド信号217の値を「0」とする。エンド信号217の値が「0」であるため、論理積ゲート出力219の値は「0」である。その結果、第2のマルチプレクサ202は、第1の加算器出力214を選択し、第2のマルチプレクサ出力213として出力する。アドレスレジスタ201は、現在格納しているアドレス、すなわちループ命令の次のアドレスを第1のアドレス10として出力する。NOP挿入信号13の値は「0」であるため、アドレスレジスタ201は第2のマルチプレクサ出力213をラッチする。

【0045】第1のアドレス10はループ命令の次のアドレスであるから、メモリ2は、ループ命令の格納されたアドレスの次のアドレスに格納された命令を、第1の命令コード11として出力する。NOP挿入信号13の値は「0」であるから、NOP生成部3に含まれる第1のマルチプレクサ102は命令レジスタ出力112を選択し、第2の命令コード12として出力する。また、命令レジスタ101は第1の命令コード11をラッチする。すなわち、NOP生成部3は、NOP命令の挿入を行わず、メモリ2から出力された第1の命令コード11の内容を順次第2の命令コード12として出力する。これは、ループ実行部1において第1の比較器207がエンド信号217の値を「1」にするまで続けられる。

【0046】ループ実行部1において、第1の比較器217は、エンドアドレス216と第1のアドレス10が一致するとエンド信号217の値を「1」とする。ループカウンタ208は、エンド信号217の値が「1」になるとそのカウンタ値を1減算する。そして、ループカウンタ208はそのカウンタ値が「0」になると、ゼロ信号218の値を「1」とする。エンド信号217の値が「1」であるとき、論理積ゲート209は、ゼロ信号210の値が「1」になると分岐信号219の値を「1」にする。分岐信号219が「1」になると、第2のマルチプレクサ202はスタートアドレス214を選択する。そのため、スタートアドレスレジスタ203の値がアドレスレジスタ201でラッチされる。その結

果、スタートアドレスが第1のアドレス10として出力されることとなる。以上の動作により、ループカウンタ208の値が「0」になるまで、アドレスレジスタ201の保持するアドレスがエンドアドレスになると、アドレスレジスタ201の保持するアドレスがスタートアドレスになることが繰り返される。第1のアドレス10にはアドレスレジスタ201が現在格納しているアドレスが出力されるため、情報処理装置において実行される命令のメモリ2上のアドレスは、エンドアドレスに到達するとスタートアドレスに分岐することとなる。この分岐が、ループカウンタ208の値が「0」になるまで繰り返される。

【0047】図6は、本発明の実施形態1によるループ処理のプログラム制御方法を示す。

【0048】ここで、ループ回数LNは、ループ命令の実行前に設定されていると仮定する。実施形態1においては、ループ命令による分岐を行うか否かの判定（以下、「分岐判定」という。）がループ命令を処理するステージのうちQ段目のステージ以降において行われるとき、NOP命令の挿入を行う場合の最大ループ対象命令数をNとし、NOP命令を挿入する回数をMとすると、 $N=Q-2$ 、 $M=N$ の関係が成立する。

【0049】以下、第1～第4の各ステップについて説明する。

（第1のステップ）ループ命令をデコードし、ループ対象命令数Lが得られる。

（第2のステップ）ループ対象命令数LがN以下であるか比較し、N以下の場合にはNOPがM回挿入される。また、ループ対象命令数LがMより大きい場合にはNOPは挿入されない。

（第3のステップ）ループ対象命令を実行する。

（第4のステップ）ループ回数LNから1を減算し、ループ回数LNが0にならなければ第3のステップに戻り、ループ回数LNが0になれば終了する。

【0050】なお、上述の例ではループ命令実行前にループ回数LNが設定されていると説明したが、ループ命令実行時にループ回数LNを設定してもよい。

【0051】図7は、本発明の実施形態1のプログラム制御方法によるループ処理の動作を示す。ここで、パイプラインステージは、IFステージ、D1ステージ、D2ステージ、EXステージの4つからなるものとする。また、分岐判定はループ命令のEXステージ（4段目のステージ）終了後に可能であるとする。 $Q=4$ であるから、 $N=Q-2$ 、 $M=N$ より、 $N=2$ 、 $M=2$ となる。

【0052】図7（a）はループ対象命令数L=1の場合の動作を示す。情報処理装置はループ命令のD1ステージにおいてループ命令をデコードする。ループ対象命令数Lが2以下なので、情報処理装置はNOP命令を2回挿入する。その後、情報処理装置は、ループ対象命令である命令1を実行する。ループ命令のEXステージ

は、命令1のIFステージに対応するから、命令1のIFステージ終了後に分岐判定が可能となる。命令1のIFステージ終了時にループ回数LNから1を減算し、ループ回数LN=0でなければ、情報処理装置は再度命令1を実行する。情報処理装置は、命令1のIFステージ終了時にLNから1を減算し、LN=0でなければ再度命令1を実行する動作をLN=0になるまで繰り返す。LN=0になると、情報処理装置はループ処理を終了し命令2を実行する。

10 【0053】なお、図7（a）において、2つのNOP命令のIFステージは点線で示されているが、これはNOP命令のIFステージが実際には存在しないことを示している。すなわち、NOP命令はメモリからフェッチする必要がないため、IFステージは不要である。そのため、D1ステージの終了後以降にしかループ命令であることは判定できないにもかかわらず、ループ命令の直後にNOP命令を挿入することが可能となっている。以下の図においても同様である。図7（b）はループ対象命令数L=2の場合の動作を示す。L=1の場合と同様にループ対象命令数Lが2以下なので、情報処理装置はループ命令のD1ステージで命令をデコードし、NOP命令を2回挿入する。その後、情報処理装置は、命令1、命令2を実行する。ループ命令のEXステージの終了後、命令2のIFステージにおいて分岐判定がされる。命令2のIFステージ終了時にループ回数LNから1を減算し、ループ回数LN=0でなければ、情報処理装置は再度命令1、命令2を実行する。情報処理装置は、命令2のIFステージ終了時にLNから1を減算し、LN=0でなければ再度命令1、命令2を実行する動作をLN=0になるまで繰り返す。LN=0になると、情報処理装置はループ処理を終了し命令2に続く命令3を実行する。

30 【0054】図7（c）はループ対象命令数L=3の場合の動作を示す。ループ対象命令数が2より大きいので、ループ命令のD1ステージで命令をデコードし、情報処理装置は、NOP命令を挿入しないで命令1、命令2、命令3を実行するループ命令のEXステージの終了後、命令3のIFステージで分岐判定がされる。命令3のIFステージ終了時にループ回数LNから1を減算し、ループ回数LN=0でなければ、情報処理装置は再度命令1、命令2、命令3を実行する。情報処理装置は、命令3のIFステージ終了時にLNから1を減算し、LN=0でなければ再度命令1、命令2、命令3を実行する動作をLN=0になるまで繰り返す。LN=0になると、情報処理装置はループ処理を終了し命令3に続く命令4を実行する。

50 【0055】以上のように本実施形態によれば、プログラム制御装置7がループ実行手段となるループ実行部1と、NOP挿入手段となるNOP生成部3と、ループ制御手段となるループ制御部4とを備えることにより、命

令格納手段であるメモリ 2 に格納されたループ命令をデコードし、デコードにより得たループ対象命令数が 2 以下の場合には NOP 命令を 2 回挿入してループ処理を実行し、ループ対象命令数が 2 より大きい場合には NOP 命令を挿入しないでループ処理を実行することができる。

【0056】（実施形態 2）次に、本発明の実施形態 2 のプログラム制御方法および装置について図面を参照して説明する。実施形態 2 において、図 1～図 3 に示されたプログラム制御装置 7 の構成はループ制御部 4 の構成を除いて実施形態 1 と共通である。

【0057】図 5 は本発明の実施形態 2 のプログラム制御装置 7 に含まれるループ制御部 4 の構成を示す。図 5 に示されたループ制御部 4 が図 4 に示されたループ制御部 4 と異なる点は、論理積ゲート 306 と、第 3 の比較器 307 を設け、レジスタ 304 の入力を論理積出力 314 から論理積出力 318 に変更した点である。図 5 において、図 4 と同一の機能を有するものには同一の符号を付してその詳細な説明を省略する。

【0058】論理積ゲート 306 は、ループ命令信号 14 と第 3 の比較器出力 317 の論理積を論理積出力 318 として出力する。第 3 の比較器 307 は、入力されたループ対象命令数 15 の値と数値「1」316 を比較し、比較の結果を第 3 の比較器出力 317 として出力する。第 3 の比較器 307 は、ループ対象命令数 15 の値が「1」より大きいときは、第 3 の比較器出力 317 の値を「0」とする。第 3 の比較器 307 は、ループ対象命令数 15 の値が「1」であるときは、第 3 の比較器出力 317 の値を「1」とする。

【0059】以下、図 1～図 3、図 5 および図 7 を参照して、ループ命令を実行する場合の本発明によるプログラム制御装置 7 の動作を説明する。なお、命令をシーケンシャルに実行する場合およびループ処理中のプログラム制御装置 7 の動作は、実施形態 1 と同様であるので省略する。また、ループ命令を実行する場合においても、NOP 挿入信号 13、ループ命令信号 14 およびループ対象命令数 15 によって制御される、ループ実行部 1 および NOP 生成部 3 の動作は、実施形態 1 と同様であるので省略する。

【0060】ループ制御部 4 に含まれるデコーダ 301 は、入力された第 2 の命令コード 12 がループ命令であると、ループ命令信号 14 を「1」にし、ループ対象命令数 15 を出力する。第 2 の比較器 302 は、入力されたループ対象命令数 15 の値が「2」以下の場合に第 2 の比較器出力 313 を「1」にする。第 3 の比較器 307 は、入力されたループ対象命令数 15 の値が「1」の場合に第 3 の比較器出力 317 を「1」にする。ループ命令信号 14 の値が「1」であるとき、論理積ゲート 303 は第 2 の比較器出力 313 が「1」になると論理積出力 314 を「1」にする。また、ループ命令信号 14

の値が「1」であるとき、論理積ゲート 306 は第 3 の比較器出力 317 が「1」になると論理積出力 318 を「1」にする。

【0061】上記より、ループ対象命令数 15 の値が「1」の場合、第 2 の比較器出力 313 および第 3 の比較器出力 317 の値はともに「1」となる。デコーダ 301 がループ命令をデコードした時点におけるループ命令信号 14 の値は「1」であるから、論理積出力 314 および論理積出力 318 の値はともに「1」となる。その結果、論理和ゲート 305 が出力する NOP 挿入信号 13 の値は「1」となり、レジスタ 304 は値「1」をラッチする。次にデコーダ 301 が NOP 命令をデコードした時点でも、レジスタ 304 の出力するレジスタ出力 315 の値は「1」であるから、論理和ゲート 305 が出力する NOP 挿入信号 13 の値は再度「1」となる。しかし、デコーダ 301 が NOP 命令をデコードした時点では、ループ命令信号 14 の値は「0」となるため、論理積出力 318 の値は「0」となる。その結果レジスタ 304 は値「0」をラッチする。従って、デコーダ 301 が 2 回目の NOP 命令をデコードする時点には、ループ命令信号 14 の値が「0」であるため論理積出力 314 の値が「0」、レジスタ 304 の出力するレジスタ出力 315 の値も「0」となる。その結果、論理和ゲート 305 が出力する NOP 挿入信号 13 の値は「0」となる。

【0062】このように、ループ対象命令数 15 の値が「1」である場合には、NOP 挿入信号 13 の値が 2 回「1」となる。その結果、NOP 生成部 3 において NOP 命令が 2 回挿入される。

【0063】ループ対象命令数 15 の値が「2」の場合、第 2 の比較器出力 313 の値は「1」となり、第 3 の比較器出力 317 の値は「0」となる。デコーダ 301 がループ命令をデコードした時点におけるループ命令信号 14 の値は「1」であるから、第 2 の比較器出力 313 の値が「1」であることより、論理積出力 314 の値は「1」となる。論理積出力 314 の値が「1」であるため、論理和ゲート 305 が出力する NOP 挿入信号 13 の値は「1」となる。一方、第 3 の比較器出力 317 の値が「0」であることより、論理積出力 318 の値は「0」となる。論理積出力 318 の値が「0」であるため、レジスタ 304 は値「0」をラッチする。従って、デコーダ 301 が次に NOP 命令をデコードする時点には、ループ命令信号 14 の値が「0」であるため論理積出力 314 の値が「0」、レジスタ 304 の出力するレジスタ出力 315 の値も「0」となる。その結果、論理和ゲート 305 が出力する NOP 挿入信号 13 の値は「0」となる。

【0064】このように、ループ対象命令数 15 の値が「2」である場合には、NOP 挿入信号 13 の値が 1 回「1」となる。その結果、NOP 生成部 3 において NO

P命令が1回挿入される。

【0065】以上のように、ループ対象命令数15の値が「1」である場合には2回、ループ対象命令数15の値が「2」である場合には1回、NOP生成部3においてNOP命令が挿入されることとなる。

【0066】図8は、本発明の実施形態2によるループ処理のプログラム制御方法を示す。

【0067】実施形態1と同様に、ループ回数LNはループ命令の実行前に設定されていると仮定する。実施形態2においては、分岐判定がループ命令を処理するステージのうちQ段目のステージ以降に行われるとき、ループ対象命令数をLとし、NOP命令の挿入を行う場合の最大ループ対象命令数をNとし、NOP命令を挿入する回数をMとすると、 $N=Q-2$ 、 $M=N-L+1$ の関係が成立する。パイプラインステージは4ステージからなり、分岐判定はループ命令のEXステージ終了後に可能であると仮定する。

【0068】以下、第1～第5の各ステップについて説明する。

(第1のステップ) ループ命令をデコードし、ループ対象命令数Lが得られる。

(第2のステップ) ループ対象命令数LがN以下であるか判断し、N以下の場合にはNOPが1回挿入される。また、ループ対象命令数LがNより大きい場合にはNOPは挿入されない。

(第3のステップ) ループ対象命令数LがN-1以下であるか判断し、N-1以下の場合にはNOPが1回挿入される。また、ループ対象命令数LがN-1より大きい場合にはNOPは挿入されない。

【0069】以下、第3のステップと同様のステップがN-2、N-3・・・のように、ループ対象命令数Lが1と比較されるまで繰り返される。

(第4のステップ) ループ対象命令を実行する。

(第5のステップ) ループ回数LNから1を減算し、ループ回数LNが0にならなければ第4のステップに戻り、ループ回数LNが0になれば終了する。

【0070】なお、ループ命令実行時にループ回数LNを設定してもよいのは、実施形態1と同様である。

【0071】図9は、本発明の実施形態2のプログラム制御方法によるループ処理の動作を示す。実施形態1と同様に、パイプラインステージは4ステージからなり、分岐判定はループ命令のEXステージ終了後に可能であると仮定する。 $Q=4$ であるから、 $N=Q-2$ 、 $M=N-L+1$ より、 $N=2$ 、 $M=3-L$ となる。

【0072】図9(a)はループ対象命令数L=1の場合の動作を示す。この場合の情報処理装置の動作は、図7(a)に示された実施形態1の場合と同様である。

【0073】図9(b)はループ対象命令数L=2の場合の動作を示す。L=1の場合と同様にループ対象命令数Lが1なので、情報処理装置はループ命令のD1ステ

ージで命令をデコードし、NOP命令を1回挿入する。その後、情報処理装置は、命令1、命令2を実行する。命令2のIFステージにおいて分岐判定がされる。命令2のIFステージ終了時にループ回数LNから1を減算し、ループ回数LN=0でなければ、情報処理装置は再度命令1、命令2を実行する。情報処理装置は、命令2のIFステージ終了時にLNから1を減算し、LN=0でなければ再度命令1、命令2を実行する動作をLN=0になるまで繰り返す。LN=0になると、情報処理装置はループ処理を終了し命令2に続く命令3を実行する。

【0074】図9(c)はループ対象命令数L=3の場合の動作を示す。この場合の情報処理装置の動作は、図7(c)に示された実施形態1の場合と同様である。

【0075】以上のように本実施形態によれば、プログラム制御装置7がループ実行手段となるループ実行部1と、NOP挿入手段となるNOP生成部3と、ループ制御手段となるループ制御部4とを備えることにより、命令格納手段であるメモリ2に格納されたループ命令をデコードし、デコードにより得たループ対象命令数が1の場合にはNOP命令を2回挿入しループ処理を実行し、ループ対象命令数が2の場合にはNOP命令を1回挿入しループ処理を実行し、ループ対象命令数が2より大きい場合にはNOP命令を挿入しないでループ処理を実行することができる。実施形態2では、実施形態1と比較して、ループ対象命令数が2の場合に挿入するNOP命令の数を1回削減することができる。

【0076】なお、上述の実施形態1および実施形態2において、情報処理装置のパイプライン段数を4として説明したが、パイプライン段数は任意の段数でよいことは明らかである。また、分岐判定はループ命令のEXステージ以降に行うことができるとしたが、分岐判定を行うことができるようになる位置は任意の位置にしてよいことは明らかである。

【0077】

【発明の効果】ループ命令をデコードすることにより得たループ対象命令数が所定の数N以下の場合にはNOP命令をM回挿入してループ処理を実行し、ループ対象命令数がNより大きい場合にはNOP命令を挿入しないでループ処理を実行するように制御することで、ループ対象命令数がN以下の場合であってもループ処理を実行可能にするとともに、ループ対象命令数がNより大きい場合には無駄なNOP命令が挿入されないようにすることが可能となる。また、ループ対象命令数がN以下の場合にNOP命令が挿入される数を削減することも可能となる。

【図面の簡単な説明】

【図1】本発明によるプログラム制御装置の構成を示す図である。

【図2】本発明によるプログラム制御装置に含まれるN

19

OP生成部の構成を示す図である。

【図3】本発明によるプログラム制御装置に含まれるループ実行部の構成を示す図である。

【図4】本発明の実施形態1のプログラム制御装置に含まれるループ制御部の構成を示す図である。

【図5】本発明の実施形態2のプログラム制御装置に含まれるループ制御部の構成を示す図である。

【図6】本発明の実施形態1によるループ処理のプログラム制御方法を示す図である。

【図7】本発明の実施形態1のプログラム制御方法によるループ処理の動作を示す図である。

【図8】本発明の実施形態2によるループ処理のプログラム制御方法を示す図である。

【図9】本発明の実施形態2のプログラム制御方法によるループ処理の動作を示す図である。

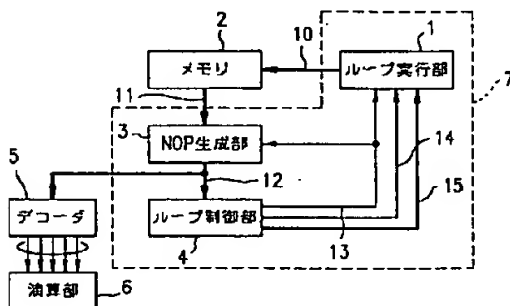
【図10】3段パイプラインの情報処理装置におけるループ処理の動作を示す図である。

【図11】4段パイプラインの情報処理装置におけるループ処理の動作を示す図である。

【符号の説明】

- 1 ループ実行部
- 2 メモリ
- 3 NOP生成部
- 4 ループ制御部
- 5 デコーダ
- 6 演算部
- 7 プログラム制御装置
- 10 第1のアドレス

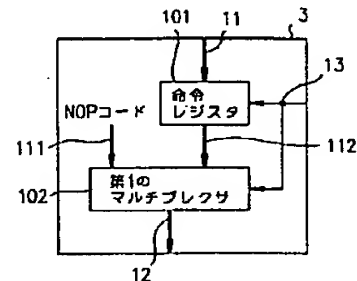
【図1】



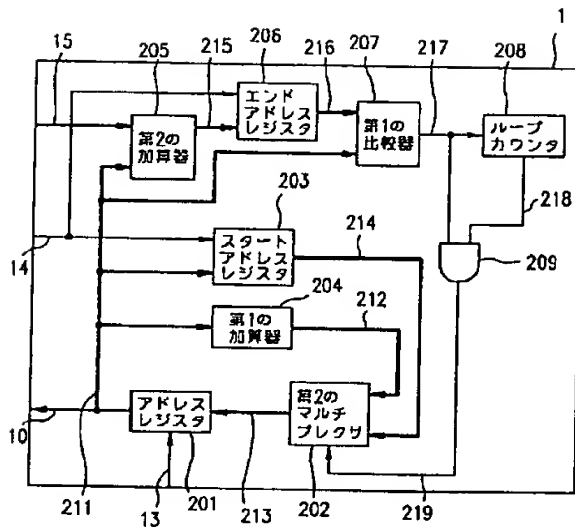
20

- 11 第1の命令コード
- 12 第2の命令コード
- 13 NOP挿入信号
- 14 ループ命令信号
- 15 ループ対象命令数
- 101 命令レジスタ
- 102 第1のマルチプレクサ
- 111 NOPコード
- 201 アドレスレジスタ
- 202 第2のマルチプレクサ
- 203 スタートアドレスレジスタ
- 204 第1の加算器
- 205 第2の加算器
- 206 エンドアドレスレジスタ
- 207 第1の比較器
- 208 ループカウンタ
- 209、303、306 論理積ゲート
- 214 スタートアドレス
- 216 エンドアドレス
- 217 エンド信号
- 218 ゼロ信号
- 219 分岐信号
- 301 デコーダ
- 302 第2の比較器
- 304 レジスタ
- 305 論理和ゲート
- 307 第3の比較器

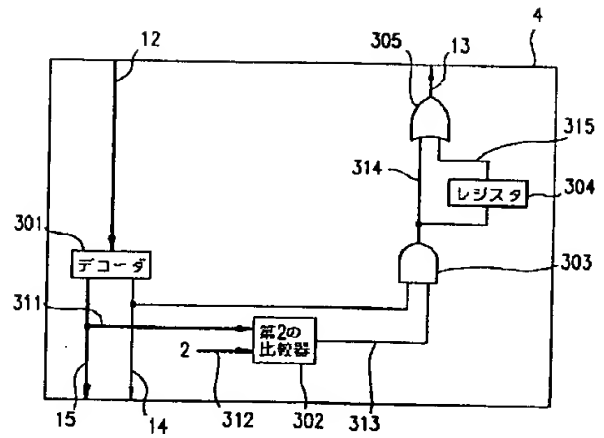
【図2】



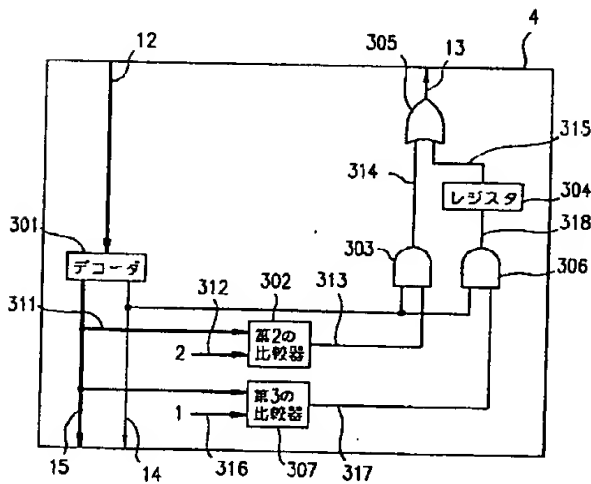
【図3】



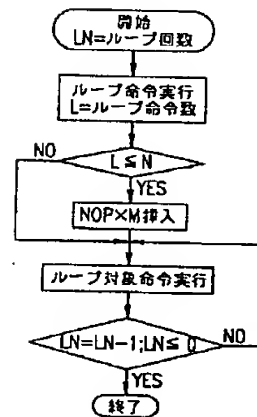
【図4】



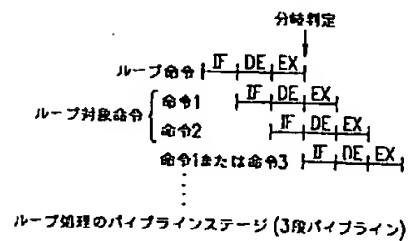
【図5】



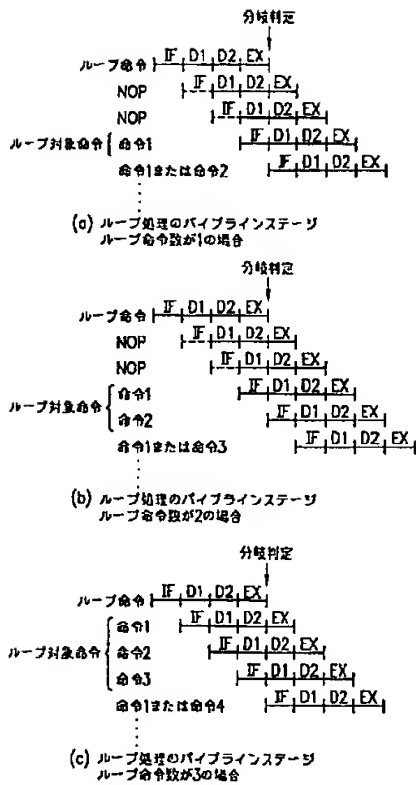
【図6】



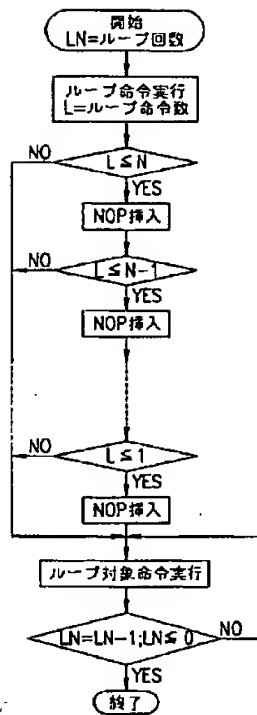
【図10】



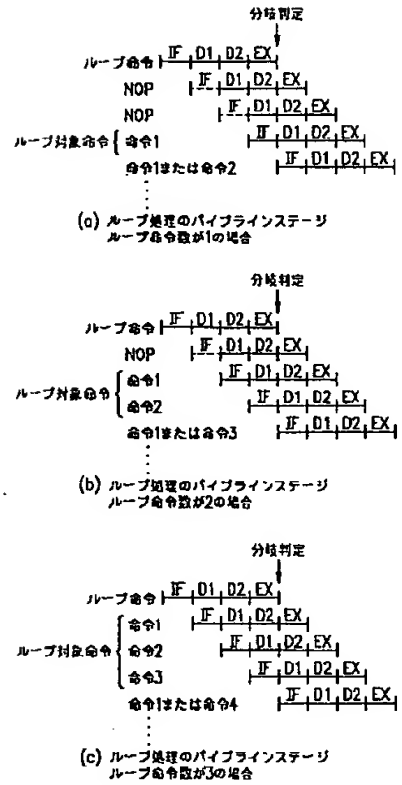
【図 7】



【図 8】



【図 9】



【図 11】

